FIGURE 1 - General Overview of Distributed File Storage System

110

120

Communication Fabric

130 131 132 133 134 135

S1 S2 S3 S4 S5 S6

151

140 141

160

150

100

DATA

110

120

211

210

FC-API

Fibre Channel

Fibre Channel

Communication with other server nodes

Client Polling 270

245 CSO

SI 240

250 FS

255 DSO

JI 260

Disk Polling 280

205

220 FC-API

Fibre Channel

Fibre Channel

221

222

Communication with other server nodes

Gigabit Ethernet

10/100T Ethernet

232

290 Admin Interface

231

230

CI 286

285

CI 291
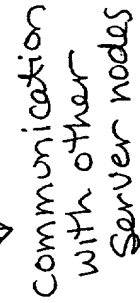
FIGURE 2 : One Embodiment of a Server Node

G-node Table ~ 330

contains information about File attributes

(See Figure 6)

points to a G-node ~ 320

Gee Table

contains information about File block location

(See Figure 5)

310

Filename Table

contains information about File names

(See Figure 4)

points to a Gee-string

points to the file's Gnode and to the Parent's G-node

points to a Child Gnid Index

points to a Gee

points to a Cache Node

Cache Node Table

contains information about cached file block

(See Figure 8)

350

points to a filename entry

Gnid Table

contains information about Directory Structure

(See Figure 7)

340

Other Servers' G-node Tables (for child files stored on other servers)

FIGURE 3 - Five metadata structures
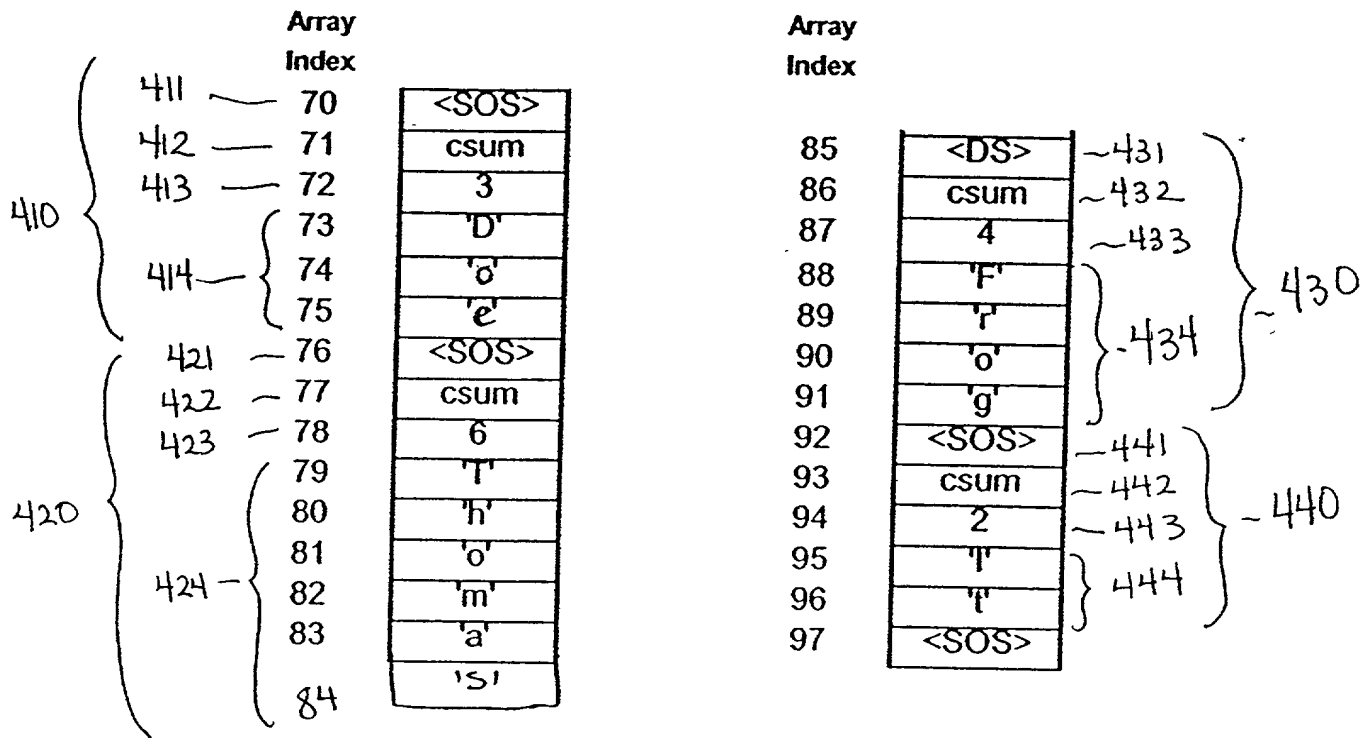
FIGURE 4 - Sample Portion of a Filename Table

320

	Index	G-Code	Data	File Logical Block
510	45	GNODE	Gnode = 67, Extent = 2, Root = TRUE	
511	46	DATA	Disk Logical Blocks: 456, 457 Drive 13	1
512	47	DATA	Disk Logical Blocks: 667, 668 Drive 15	2
513	48	DATA	Disk Logical Blocks: 112, 113 Drive 19	3
514	49	PARITY	Disk Logical Blocks: 554, 555 Drive 2	
515	50	DATA	Disk Logical Blocks: 458, 459 Drive 13	4
516	51	DATA	Disk Logical Blocks: 669, 670 Drive 15	5
517	52	DATA	Disk Logical Blocks: 119, 120 Drive 19	6
518	53	PARITY	Disk Logical Blocks: 556, 557 Drive 2	
519	54	LINK	Index 76	
	...	...	...	
520	76	GNODE	Gnode = 67, Extent = 3, Root = FALSE	
521	77	DATA	Disk Logical Blocks: 460, 461, 462 Drive 13	7
522	78	DATA	Disk Logical Blocks: 671, 672, 673 Drive 15	8
523	79	PARITY	Disk Logical Blocks: 121, 122, 123 Drive 19	
524	80	LINK	Index 88	
	...	...	...	
525	88	GNODE	Gnode = 67, Extent = 3, Root = FALSE	
526	89	DATA	Disk Logical Blocks: 463, 464, 465 Drive 13	9
527	90	DATA	Disk Logical Blocks: 674, 675, 676 Drive 15	10
528	91	PARITY	Disk Logical Blocks: 124, 125, 126 Drive 19	
529	92	GNODE	Gnode = 43, Extent = 4, Root = FALSE	
	...	...	...	

590   591   592   550   551   552   500

FIGURE 5 · Sample Portion of a Gee Table

Attribute Data
Attribute Data
602 — File Attribute – type
604 — File Attribute – mode
606 — File Attribute – links
608 — File Attribute – uid
610 — File Attribute – gid
612 — File Attribute – size
614 — File Attribute – used
620 — File Attribute – fileId
622 — File Attribute – atime
624 — File Attribute – mtime
626 — File Attribute – ctime
628 — Child Gnid Index
630 — Gee Index – Last Used
631 — Gee Offset – Last Used
632 — Gee Index – Midpoint
633 — Gee Offset – Midpoint
634 — Gee Index – Tail
635 — Gee Offset – Tail
636 — Gee Index – Root
638 — Gnode Status
640 — Quick Shot Status
642 — Quick Shot Link

600

FIGURE 6 — G-NODE ATTRIBUTES

700

Status	Status	Status	Status
Parent Gnode Ptr	Parent Gnode Ptr	Parent Gnode Ptr	Parent Gnode Ptr
Sibling Gnid Ptr →	Sibling Gnid Ptr →	Sibling Gnid Ptr →	Sibling Gnid Ptr
Gnode Ptr	Gnode Ptr	Gnode Ptr	Gnode Ptr
Filename Ptr	Filename Ptr	Filename Ptr	Filename Ptr
ForBiGnid Ptr	ForBiGnid Ptr	ForBiGnid Ptr	ForBiGnid Ptr
BckBiGnid Ptr	BckBiGnid Ptr	BckBiGnid Ptr	BckBiGnid Ptr

720~ 730~ 740~ 750~ 760 770 780

710   711   712   713

FIGURE 7 - Structure of a Gnid String

DataGee
PrevPtr
NextPtr
CacheBlockAddr
ReadCt
CacheTime
Regenerated
CacheBlockHiAddr
CacheBlockLoAddr

810 —
815 —
820 —
825 —
830 —
835 —
840 —
845 —
850 —

} 800

FIGURE 8a - Structure of a Cache Node

350

Normal List

Write List Cache Nodes are returned to "top" of Normal List

Most recently used

875

Read List cache nodes are returned to 'top' of Normal List

Write List

Read List

Normal List Cache Nodes may come from anywhere on Normal List

Normal List Cache Nodes may come from anywhere on Normal List

Least recently used

880

860

870

865
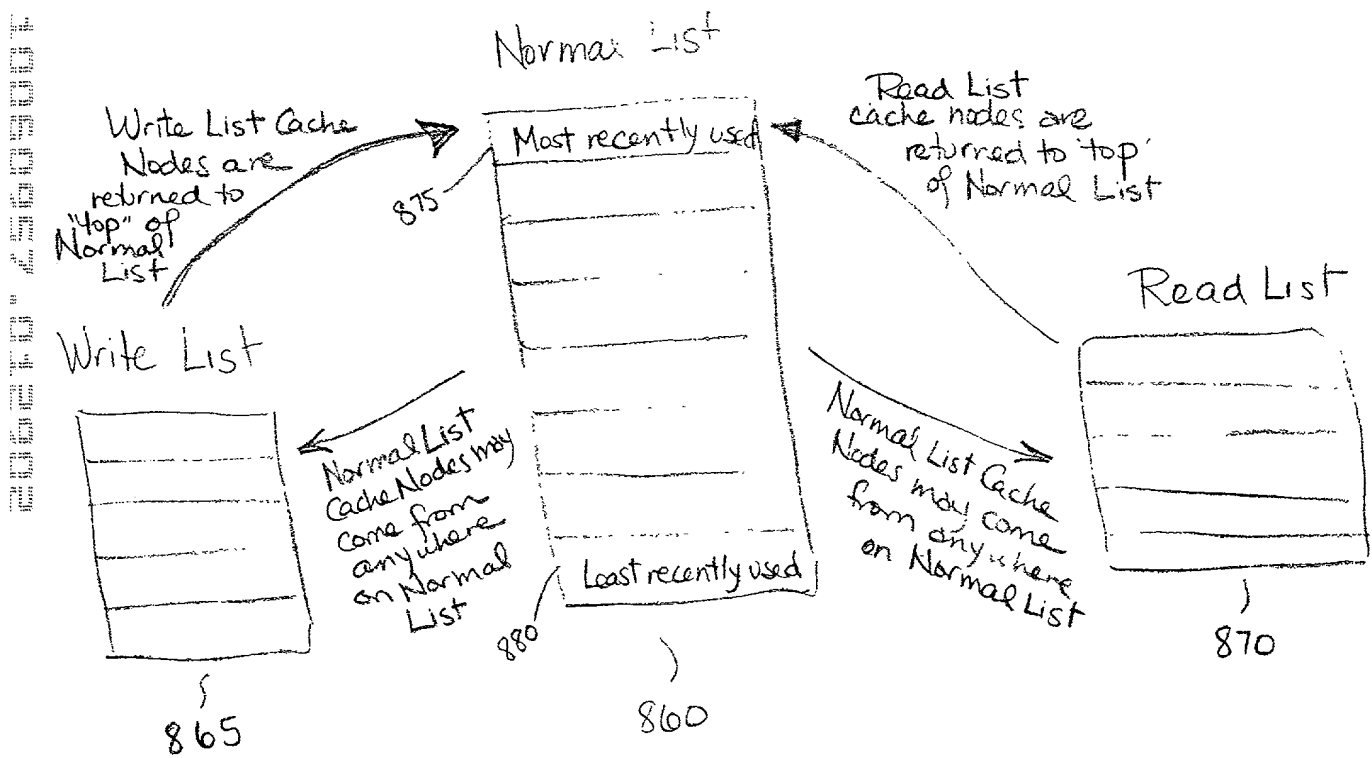
FIGURE 8B – Conceptual division of a Cache Node Table into Three Lists

900

911 912 921 922 923

930	lockStatus
931	timeoutCt
932	next
933	pending
934	mode
935	access
936	clientId
937	uid

	lockStatus
	timeoutCt
	next
	pending
	mode
	access
	clientId
	uid

	lockStatus
	timeoutCt
	next
938	pending
939	offset
940	length
941	exclusive
942	clientId
943	uid
	svid

	lockStatus
	timeoutCt
	next
	pending
	offset
	length
	exclusive
	clientId
	uid
	svid

	lockStatus
	timeoutCt
	next
	pending
	offset
	length
	exclusive
	clientId
	uid
	svid

910

920

FIGURE 9 - A Sample Lock String

1020
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1010
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1030
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1000
1001 clientId
1002 counter
1003 ltLink
1004 gtLink
stackNext
stackPrev

1040
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1050
clientId
counter
ltLink
gtLink
stackNext
stackPrev

FIGURE 10 · Refresh Nodes configured as a binary tree.

1100
clientId
counter
ltLink
gtLink
1105 stackNext
1106 stackPrev

1110
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1120
clientId
counter
ltLink
gtLink
stackNext
stackPrev

1130
clientId
counter
ltLink
gtLink
stackNext
stackPrev

FIGURE 11 - RefreshNodes configured as a doubly-linked list

1200

1210 ~	Status
1220 ~	intent Type
1230 ~	goal Buffer Index
1240 ~	spare
1250 ~	driveSector
1260 ~	drive
1270 ~	intentData

FIGURE 12 - Structure of an Intent Log Entry

1300

1310	1320	1330
Recommended NIC	ServerID of the file's Primary owner	Gnode Index
BIT 31	BIT 29	BIT 23 .... BIT Ø

FIGURE 13 - Structure of a File Handle

CLIENT 110

1440

1420
1300
"AAAAA"
123 | 1
1320 | 1330

SERVER "123" 130

Gnode Table 330

1430 | 602 | 628 | 636

0				
1	NFDIR		3	
2				
⋮				
9	NFREG			2

1432
1448

Gnid Table 346

740 | 750 | 760

0		
1		
2		
3	4	1
4	6	0
5		
6	123,9	2
⋮		

1436
1440
1444

Filename Table 310

0	"DE"	~1442
1	"ABCD"	~1438
2	"AAAAA"	~1446
3		
⋮		

Gee Table 320

590 | 591

0		
1		
2	GNODE	Gnode=9, Root=T
3	CACHE DATA	7
4		
⋮		

1450
1452

Cache Node Table 350

1460

⋮			
7	W		

1454

Cache 1456

⋮		
W	FILE DATA	1458

SERVER "X" 132

FIGURE 14a: Example of a File Look-Up

FIGURE 14b Example of a File Access

START

1500

1510 Does Client have desired file's file handle ?

No → 1515 Client and Server Perform File Handle Look-up (See Figure 16)

YES

Client sends File Access Request to Server Indicated in File Handle 1520

Server Accesses Gnode Indicated in File Handle 1525

Server Accesses Appropriate Gee Indicated in Gnode 1530

1535 Is G-code CACHE PARITY or CACHE DATA ?

No → 1540 Server Caches File Data from Disk (See Figure 17)

YES 1545

Server Accesses Cache Node Indicated in Gee

Server Manipulates Cache Nodes as Needed 1550

Server Accesses File Data in Cache Location Indicated in Cache Node 1555

Server Performs Desired Operation on File Data 1560

END 1570

FIGURE 15: Performing a File Access

START ~1605

Client sends Look-Up Request to Server ~1610

Server Accesses Gnode Indicated in Look-Up Request ~1615

Server Uses Child Gnid Index in Gnode to Access the Directory's Gnid-String ~1620

Server Calculates Checksum and Filename Length for Desired next Pathname Component ~1625

Server Uses Filename Ptr field in Gnid to access Filename Entry ~1630

Is checksum from Filename Entry greater than checksum from Look Up Request ? ~1635

YES → Return error message to Client: File Not Found ~1640

Do checksums and filename lengths match ? ~1645

YES → Do filenames match ? ~1650

YES → Return File Handle to Client ~1655

END. File Handle Look-Up ~1665

NO → Access next Gnid on Gnid-String

NO ~1660

1600

FIGURE 16: Performing a File Handle Look-Up

1705 ~ (START)

1710 ~ | Identify Least Recently Used Cache Node |

1720 ~ | Write associated file data from cache to disk |

1730 ~ | Return Data Gee from Cache Node to Gee Table |

1740 ~ | Copy Data Gee for new data into Cache Node |

1750 ~ | Copy new data from disk to Cache |

1760 ~ (END)

1540

FIGURE 17: Caching File Data

START ~1805

Client sends file allocation request including parent directory's file handle and filename of desired file
1810

Server indicated in parent's file handle receives request
1815

Parent server uses workload statistics from other server nodes to decide: Will "parent" server own the new file?
1820

NO

"Parent" Server sends request for file allocation to a "second" server
1850

"Second" server allocates new Gnode and Filename Entry and creates a new file
1855

"Second" server sends file handle for new file back to "parent" server
1860

YES

"Parent" Server allocates new Gnode and creates new file
1830

"Parent" Server creates new Gnid and Filename Entry
1835

"Parent" Server sends file handle for new file to requesting client
1840

END
1845

FIGURE 18 - File Allocation

1800

1900

- Gnode
  Redirectors
  (GNR) 1910

gnr[n]	→ Not In Cache
...	
gnr[11]	→ Not In Cache
gnr[10]	
gnr[9]	→ Not In Cache
gnr[8]	→ Not In Cache
gnr[7]	→ Not In Cache
gnr[6]	→ Not In Cache
gnr[5]	
gnr[4]	→ Not In Cache
gnr[3]	
gnr[2]	
gnr[1]	→ Not In Cahce
gnr[0]	→ Not In Cache

File Handle is this Index

1920

| cached gnode - C |
| ... |
| cached gnode - 2 |
| cached gnode - 1 |
| cached gnode - 0 |

600

FIGURE 19

2000

2010 STATUS

2020 LINKING INFORMATION

2030 GNODE

128 Bytes

2040 File location data

16 KBytes

Figure 20a

FIGURE 20b

CONVENTIONAL RAID MAPPING
(PRIOR ART)

DISK ARRAY

DISK 0 — 2115
DISK 1
DISK 2
DISK 3
DISK 4
DISK 5 — 2110
DISK 6
DISK 7
DISK 8
DISK 9
DISK 10
DISK 11

DRIVE #

2130

	0	1	2	3	4	5	6	7	8	9	10	Parity
0												
1												
2													STRIPE 2
3													STRIPE 3
4													STRIPE 4
5												
6												
7												
8												
9												
. .												
N												

2120

STRIPE

FIGURE 21

FIGURE 22A

CLIENT   CLIENT   CLIENT   CLIENT   110

100

COMMUNICATION FABRIC   120

SERVER   130

150

140

DISK ARRAY

FIGURE 22B

CLUSTER

SERVER   SERVER

DISK ARRAY

DISK ARRAY

160

SERVER   SERVER

DISK ARRAY

DISK ARRAY

160

SERVER   SERVER

DISK ARRAY

DISK ARRAY

160

CLIENT   CLIENT   CLIENT   CLIENT   110

100

COMMUNICATION FABRIC   120

130

SERVER   SERVER   131

150

140   165   141

151

DISK ARRAY   DISK ARRAY

160

DISK ARRAY

FIGURE 23

FIGURE 24A

2400

CLIENT STORAGE
REQUEST ~2410

| DATA ~2415

RECEIVE DATA FROM
CLIENT ~2420

ANALYZE DATA /
DETERMINE BLOCK
DISTRIBUTION ~2430

PARSE INTO DATA
BLOCK(S) ~2440

DETERMINE PARITY
GROUP ARRANGEMENT ~2450

CALCULATE PARITY
BLOCKS ~2460

DETERMINE BLOCK
DISTRIBUTION FOR
STORAGE ~2470

STORE BLOCKS ~2480

2405

FIGURE 24B

CLIENT STORAGE
REQUEST ⌐2410

ANALYZE DATA /
DETERMINE BLOCK
DISTRIBUTION ⌐2430

PARSE INTO DATA
BLOCK(S) ⌐2440

DETERMINE PARITY
GROUP ARRANGEMENT ⌐2450

DETERMINE BLOCK
DISTRIBUTION FOR
STORAGE ⌐2470

DATA ⌐2415

RECEIVE DATA FROM
CLIENT ⌐2420

CALCULATE PARITY
BLOCKS ⌐2460

STORE BLOCKS ⌐2480

FIGURE 25

FIGURE 26A

PARITY GROUP A (BLOCK NUMBER= 4)

2330 → 2335 → 2337 → 2502

| DATA 1₁ | → | DATA 2₁ | → | DATA 3₁ | → | DATA 4₁ | - - - | PARITY1-4 ₁ |

PARITY GROUP B (BLOCK NUMBER= 2)

2337 → 2504

| DATA 5₁ | → | DATA 6₁ | - - - | PARITY 5-6 ₁ |

2330   2505

FIGURE 26B

PARITY GROUP C
(BLOCK NUMBER = 4 / BLOCK SIZE = 256K)

2335   2335

2330 → 2337 → 2506

| DATA 1₁ | → | DATA 2₁ | → | DATA 3₁ | → | DATA 4₁ | - - - | PARITY1-4 ₁ |

PARITY GROUP C
(BLOCK NUMBER = 4 / BLOCK SIZE = 128K)

2335

2330   2505   2337 → 2508

| DATA 1₁ | → | DATA 2₁ | → | DATA 3₁ | → | DATA 4₁ | - - - | PARITY1-4 ₁ |

DISK ARRAY INITIALIZATION USING GEE TABLE
SPACE ALLOCATION

2530

INDEX	G-CODE	DATA
...	...	...
45	GNODE	EXTENT=2
46	DATA	BLOCKS 456, 457: Drive 13
47	DATA	BLOCKS 667, 668: Drive 15
48	DATA	BLOCKS 112, 113: Drive 19
49	PARITY	BLOCKS 554, 555: Drive 2
...	...	...
76	GNODE	EXTENT=3
77	DATA	BLOCKS 460, 461, 462: Drive 13
78	DATA	BLOCKS 671, 672, 673: Drive 15
79	PARITY	BLOCKS 121, 122, 123: Drive 19
...	...	...
88	GNODE	EXTENT=2
89	DATA	BLOCKS 463, 464, 465: Drive 2
90	DATA	BLOCKS 674, 675, 676: Drive 5
91	PARITY	BLOCKS 124, 125, 126: Drive 13
...		

2532   2534   2536

2542

2538

2538

2540

2540

2540

FIGURE 27

ARRAY PREPARATION / G-TABLE FORMATTING

2448

2550 — DETERMINE DISK
CHARACTERISTICS

2555 — DETERMINE BLOCK / PARITY
GROUP ALLOTMENT

2560 — DETERMINE MAPPING / CREATE
GEE TABLE

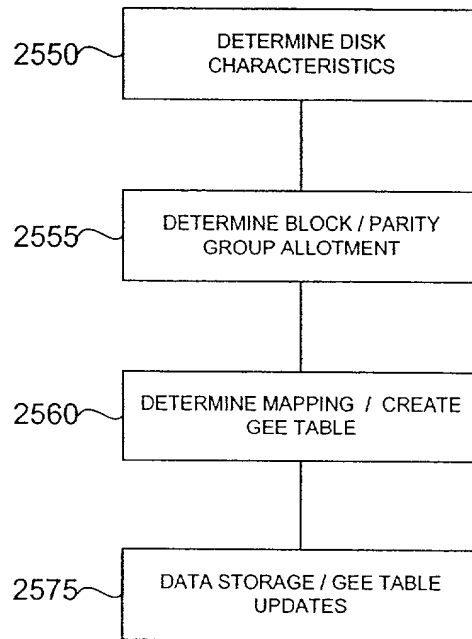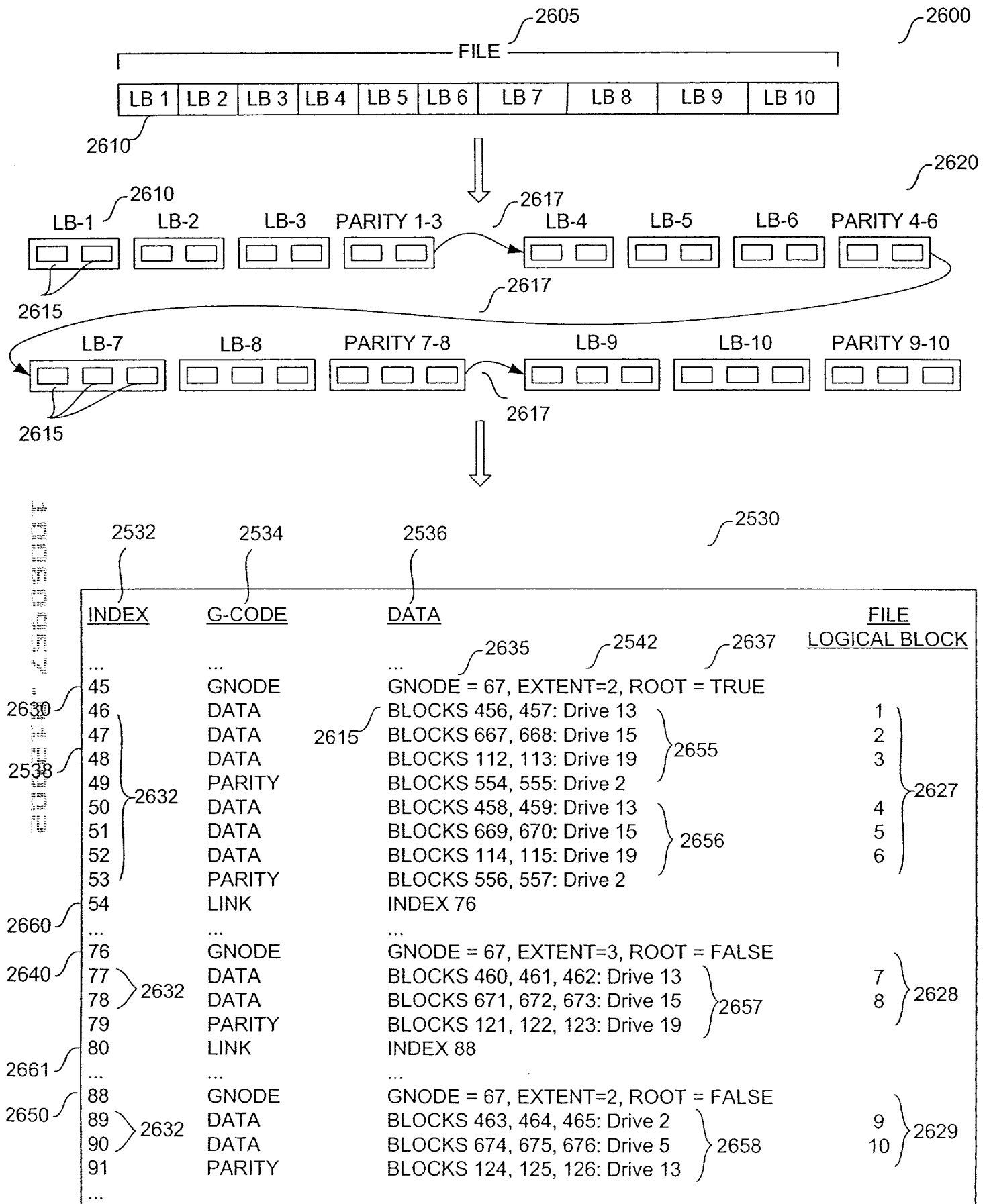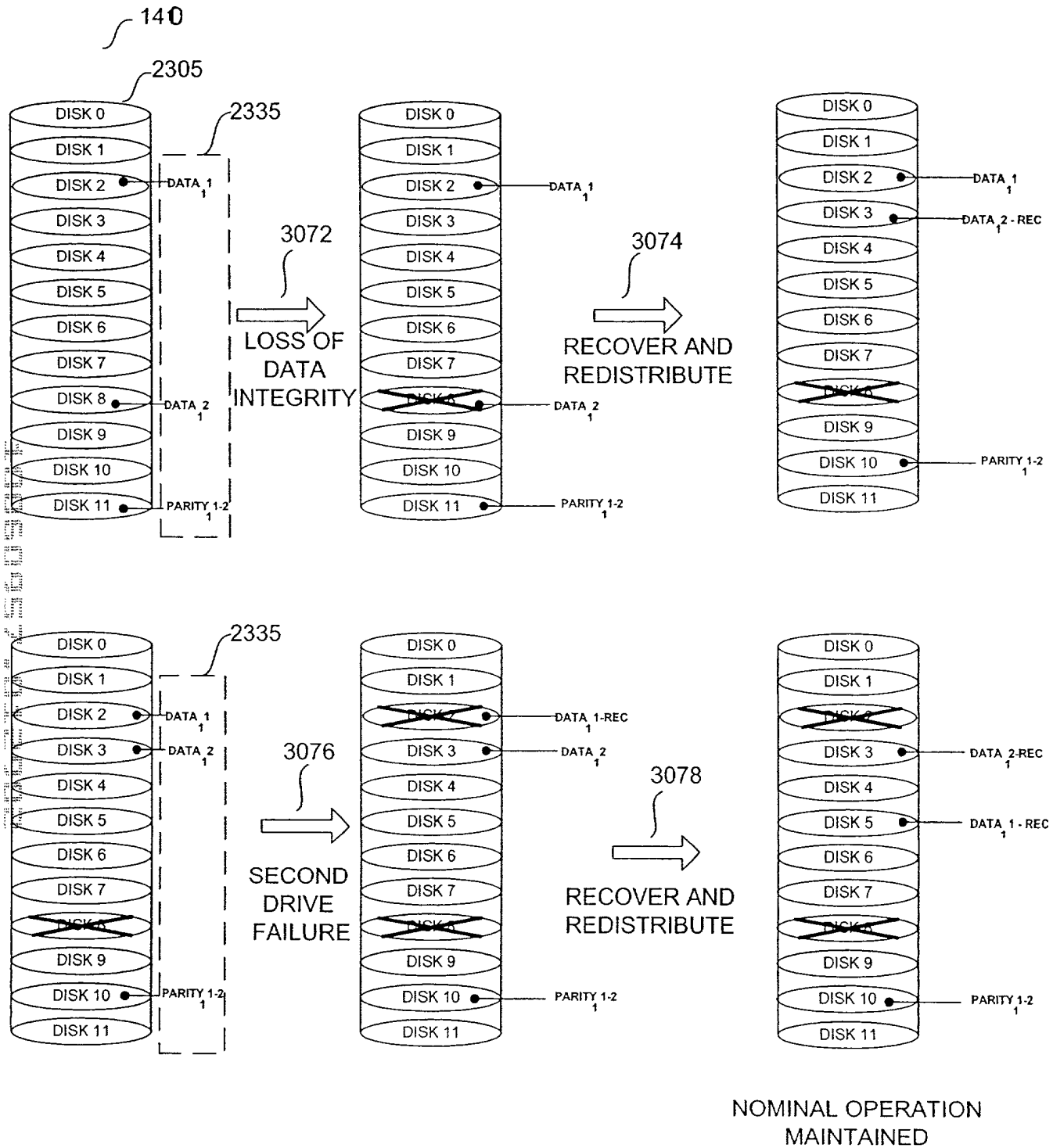2575 — DATA STORAGE / GEE TABLE
UPDATES

FIGURE 28

FIGURE 29

DRIVE FAILURE RECOVERY MECHANISM

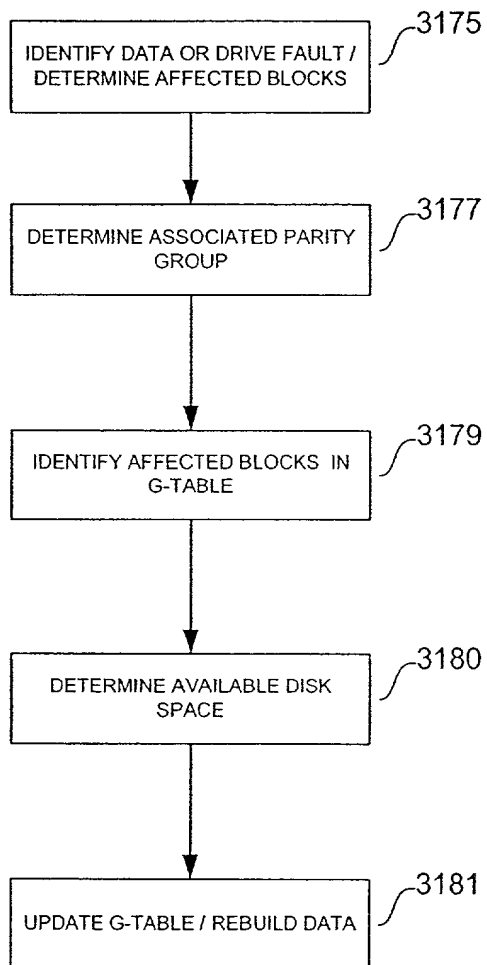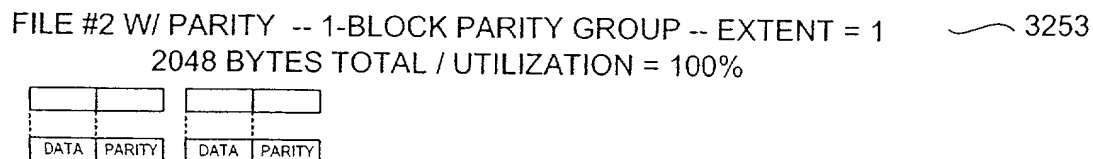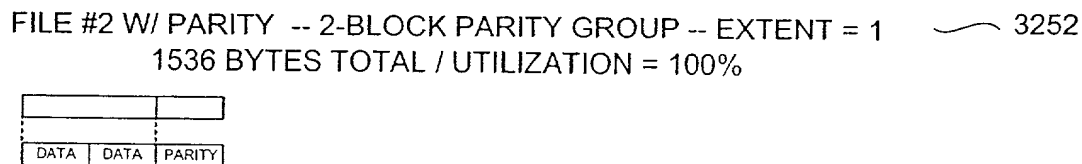FIGURE 30

DATA RECOVERY
PROCESS

3172

IDENTIFY DATA OR DRIVE FAULT /
DETERMINE AFFECTED BLOCKS

3175

DETERMINE ASSOCIATED PARITY
GROUP

3177

IDENTIFY AFFECTED BLOCKS IN
G-TABLE

3179

DETERMINE AVAILABLE DISK
SPACE

3180

UPDATE G-TABLE / REBUILD DATA

3181

FIGURE 31

FILE #1

0   4096

FILE #1 W/ PARITY -- 4-BLOCK PARITY GROUP -- EXTENT = 2   ~ 3240
5120 BYTES TOTAL / UTILIZATION = 100%
~3245

| DATA | DATA | DATA | DATA | PARITY |

0   4096

FILE #1 W/ PARITY -- 3-BLOCK PARITY GROUP -- EXTENT = 2   ~ 3241
8192 BYTES TOTAL / UTILIZATION = 66%

~3247   ~3246

| DATA | DATA | DATA | PARITY | | DATA | UNUSED | UNUSED | PARITY |

FILE #1 W/ PARITY -- 2-BLOCK PARITY GROUP -- EXTENT = 1   ~ 3242
6144 BYTES TOTAL / UTILIZATION = 100%

| DATA | DATA | PARITY | | DATA | DATA | PARITY | | DATA | DATA | PARITY | | DATA | DATA | PARITY |

FILE #1 W/ PARITY -- 1-BLOCK PARITY GROUP -- EXTENT = 1   ~ 3243
8192 BYTES TOTAL / UTILIZATION = 100%

| DATA | PARITY | | DATA | PARITY | | DATA | PARITY | | DATA | PARITY | | DATA | PARITY | | DATA | PARITY | | DATA | PARITY | | DATA | PARITY |

FILE #2

0   1024

FILE #2 W/ PARITY -- 4-BLOCK PARITY GROUP -- EXTENT = 2   ~ 3250
5120 BYTES TOTAL / UTILIZATION = 25%
3245

| UNUSED | UNUSED | UNUSED | DATA | PARITY |

FILE #2 W/ PARITY -- 3-BLOCK PARITY GROUP -- EXTENT = 2   ~ 3251
4096 BYTES TOTAL / UTILIZATION = 33%

| UNUSED | UNUSED | DATA | PARITY |

FILE #2 W/ PARITY -- 2-BLOCK PARITY GROUP -- EXTENT = 1   ~ 3252
1536 BYTES TOTAL / UTILIZATION = 100%

| DATA | DATA | PARITY |

FILE #2 W/ PARITY -- 1-BLOCK PARITY GROUP -- EXTENT = 1   ~ 3253
2048 BYTES TOTAL / UTILIZATION = 100%

| DATA | PARITY | | DATA | PARITY |

```
                                          ⌒ 3360

                    ┌─────────────────────┐  ⌒ 3361
                    │  RECEIVE FILE DATA   │
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐  ⌒ 3363
                    │   ASSES FILE DATA    │
                    │   CHARACTERISTICS    │
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐  ⌒ 3365
                    │ IDENTIFY AVAILABLE PARITY │
                    │  GROUP CONFIGURATIONS │
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐  ⌒ 3367
                    │  SELECT PARITY GROUP │
                    │ CONFIGURATION BASED ON │
                    │ UTILIZATION CHARACTERISTICS │
                    └─────────────────────┘
                              │
                              ▼
                    ┌─────────────────────┐  ⌒ 3369
                    │   STORE FILE DATA IN │
                    │ SELECTED PARITY GROUP │
                    └─────────────────────┘


                         FIGURE 33
```

FIGURE 34A

INITIAL ALLOCATION ⌐3491   DISK SPACE % ⌐3485

		INITIAL ALLOCATION	DISK SPACE %
	DATA	DATA	DATA	DATA	PARITY	4 block parity ⌐3480	10000 groups	36%
DATA	DATA	DATA	PARITY	3 block parity ⌐3481	10000 groups	28%
DATA	DATA	PARITY	2 block parity ⌐3482	10000 groups	22%
DATA	PARITY	1 block parity ⌐3483	10000 groups	14%

DISK USAGE ⌐3487

FIGURE 34B

	3492 FREE	OCCUPIED 3490	TOTAL	DISK SPACE %
3480 4 block parity	2500 groups	7500 groups	10000 groups	36%
3481 3 block parity	7500 groups	2500 groups	10000 groups	28%
3482 2 block parity	3500 groups	6500 groups	10000 groups	22%
3483 1 block parity	500 groups	9500 groups	10000 groups	14%

REDISTRIBUTION ⌐3494

FIGURE 34C

		3492 FREE	OCCUPIED 3490	TOTAL	DISK SPACE %
3480 4 block parity		2500 groups	7500 groups	10000 groups	36%
3481 3 block parity	-5000 groups of 3 block parity	2500 groups	2500 groups	5000 groups	14%
3482 2 block parity		3500 groups	6500 groups	10000 groups	22%
3483 1 block parity	+10000 groups of 1 block parity	10500 groups	9500 groups	20000 groups	28%

REDISTRIBUTION

PARITY GROUP REDISTRIBUTION PROCESSES ⨍ 3500

⨍ 3510

FIGURE 35A   PARITY GROUP DISSOLUTION

5-BLOCK PARITY GROUP ⨍ 3515

DATA	DATA	DATA	DATA	DATA	PARITY

1-BLOCK PARITY ⨍ 3520 GROUP

DATA	PARITY

3-BLOCK PARITY ⨍ 3525 GROUP

DATA	DATA	DATA	PARITY

OR

2-BLOCK PARITY ⨍ 3530 GROUP

DATA	DATA	PARITY

2-BLOCK PARITY ⨍ 3530 GROUP

DATA	DATA	PARITY

OR

1-BLOCK PARITY ⨍ 3520 GROUP

DATA	PARITY

1-BLOCK PARITY ⨍ 3520 GROUP

DATA	PARITY

1-BLOCK PARITY ⨍ 3520 GROUP

DATA	PARITY

⨍ 3535
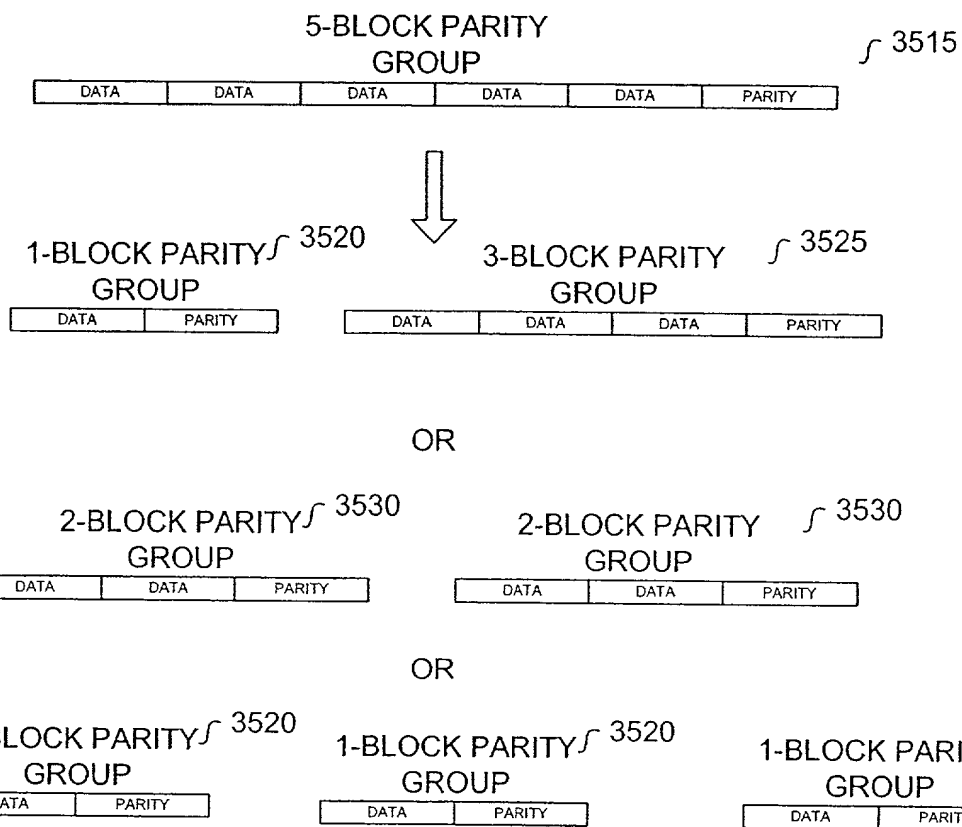
FIGURE 35B

PARITY GROUP CONSOLIDATION

3-BLOCK PARITY GROUP ⨍ 3525

DATA	DATA	DATA	PARITY

2-BLOCK PARITY GROUPS ⨍ 3530

DATA	DATA	PARITY

DATA	DATA	PARITY

1-BLOCK PARITY ⨍ 3520 GROUP

DATA	PARITY

OR

5-BLOCK PARITY GROUP ⨍ 3515

DATA	DATA	DATA	DATA	DATA	PARITY

3600

3602

MONITOR PARITY GROUP
AVAILABILITY

NO

3604

PARITY GROUP
DEPLETION
OBSERVED
?

YES

3606

ASSESS PARITY GROUP
STATISTICS

3608

REDISTRIBUTE PARITY GROUPS

FIGURE 36

3492

REMAINING
FREE /   3490 3492
AVAILABLE   OCCUPIED   REMAINING   3490
                                  FREE / AVAILABLE   OCCUPIED

3480
  4 block parity   2800 groups 7200 groups   5400 groups   9200 groups
3481   +2000 groups
  3 block parity   1800 groups 8200 groups ⟹ 1800 groups   of 4-block parity   8200 groups

3482
  2 block parity   800 groups 9200 groups OPTIMIZATION 800 groups   -3200 groups 9200 groups
3483   )   of 1-block parity
  1 block parity   2300 groups 7700 groups   3785   300 groups   4500 groups

                                                                    Equivalent space
                                                                    of 1200 groups
                                                      1200   of 1-block parity
                                                                          freed

FIGURE 37

3800

3851

PERFORM FILE OPERATIONS

3854

COLLECT RESOURCE UTILIZATION
STATISTICS

3852

COLLECT FILE ACCESS STATISTICS

3856

PREDICT FUTURE FILE AND
RESOURCE UTILIZATION
(WORKLOAD)

3858
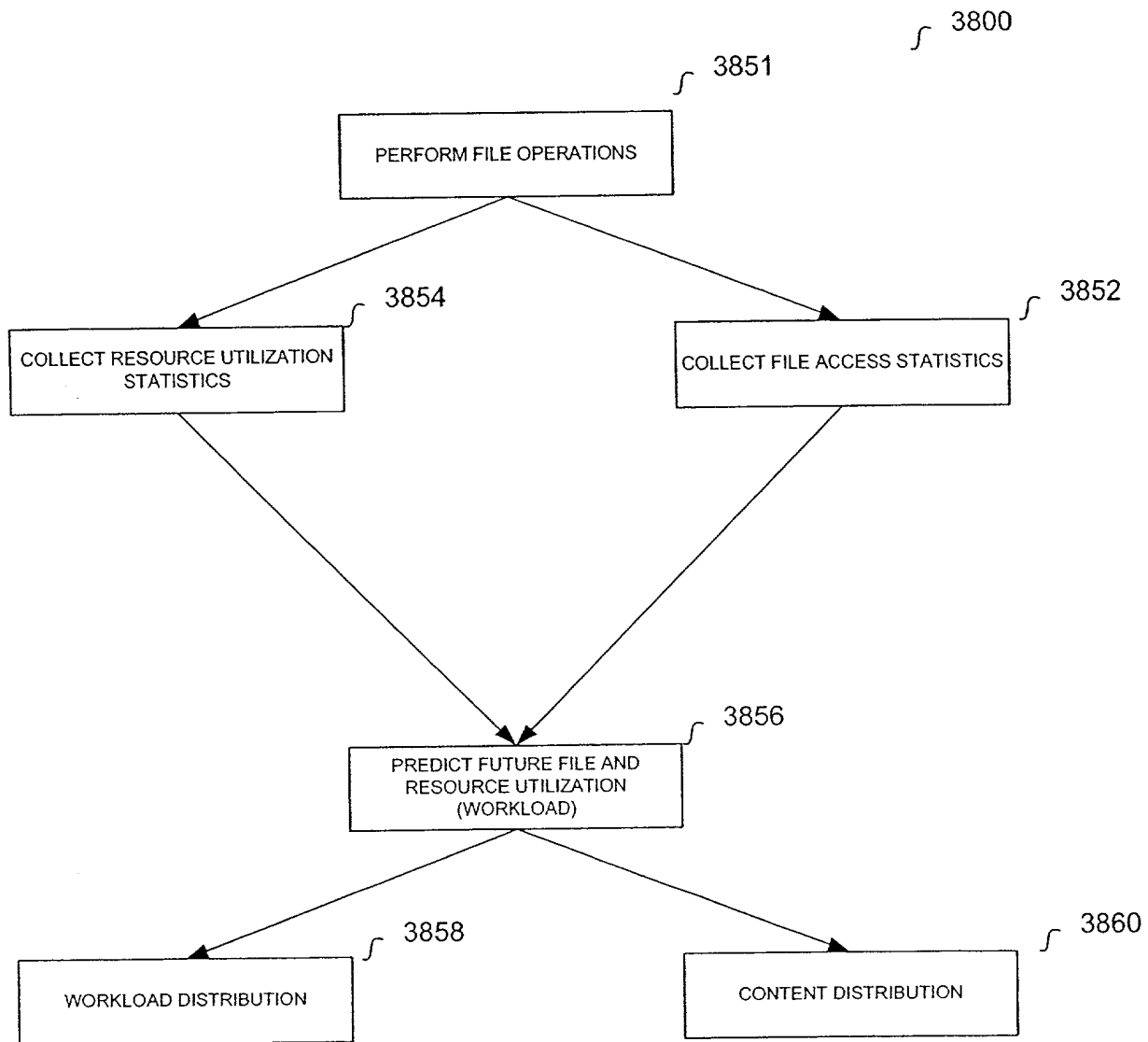
WORKLOAD DISTRIBUTION

3860

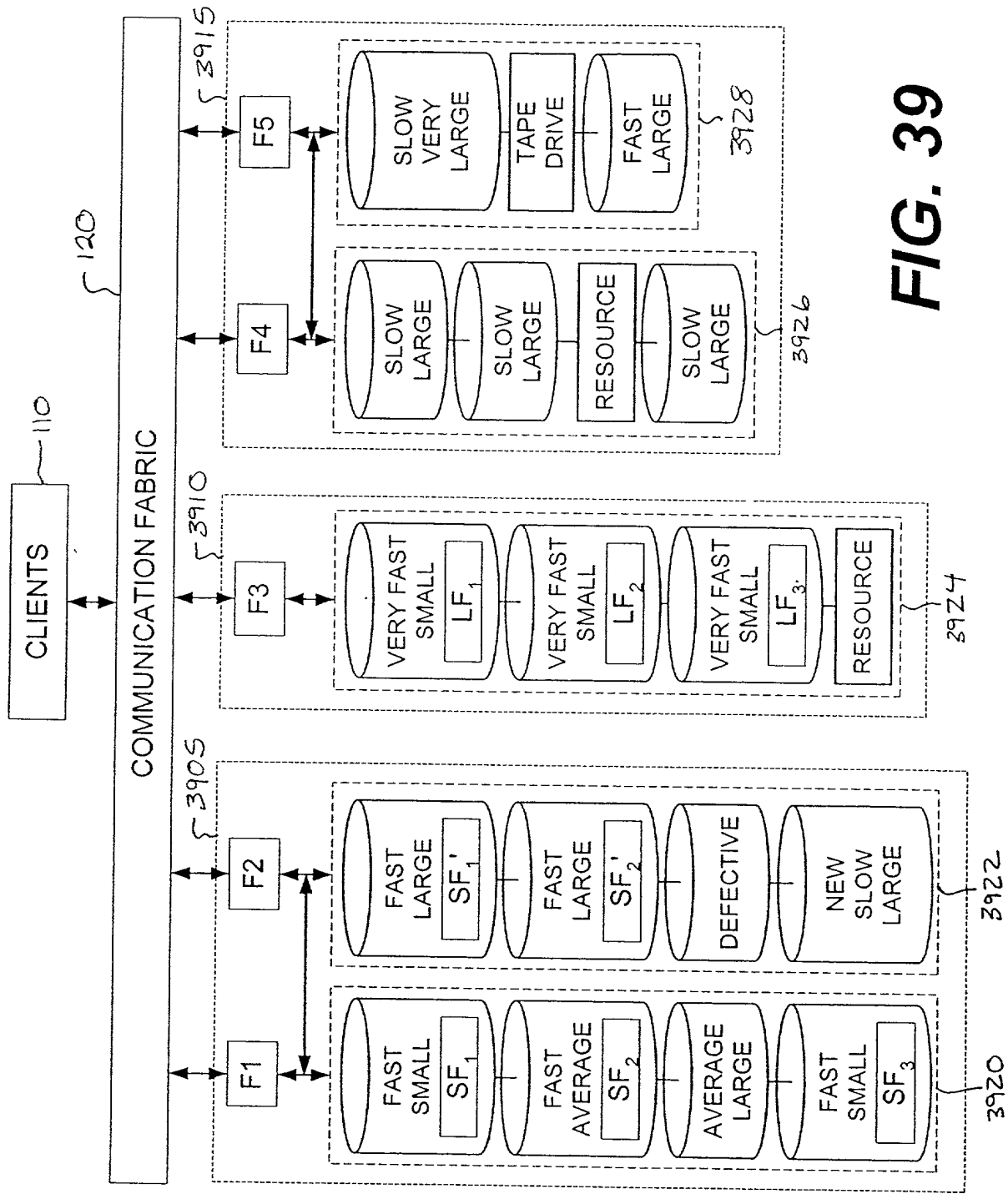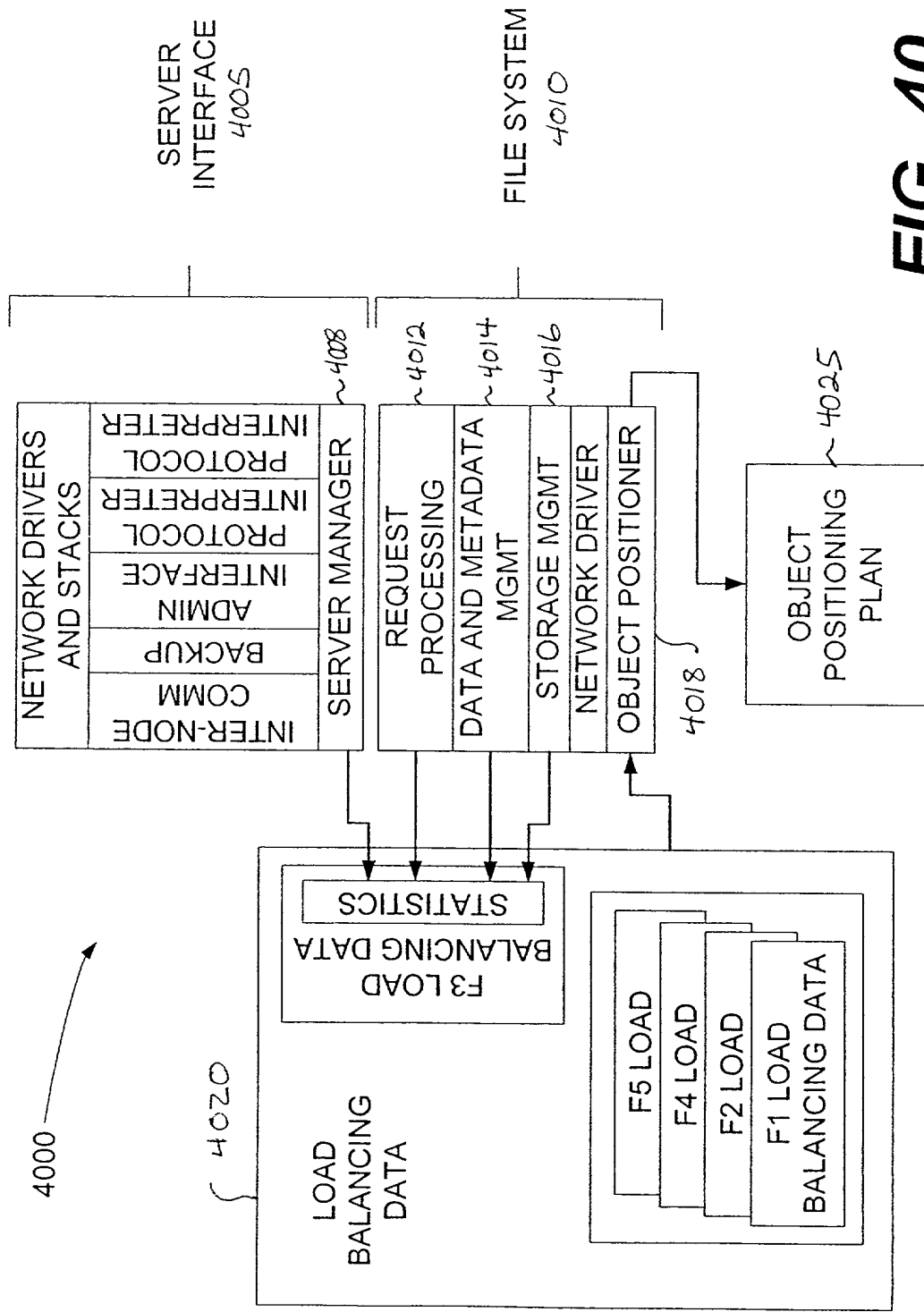CONTENT DISTRIBUTION

FIGURE 38

FIG. 39

FIG. 40

F3 OBJECT
POSITIONING PLAN

-Push LF to F4-F5 Cluster

-Issue File Handle For LF = Stale

-If Requested,
  -Send acceptance for copy
    of SF to F1
  -Create copy of SF
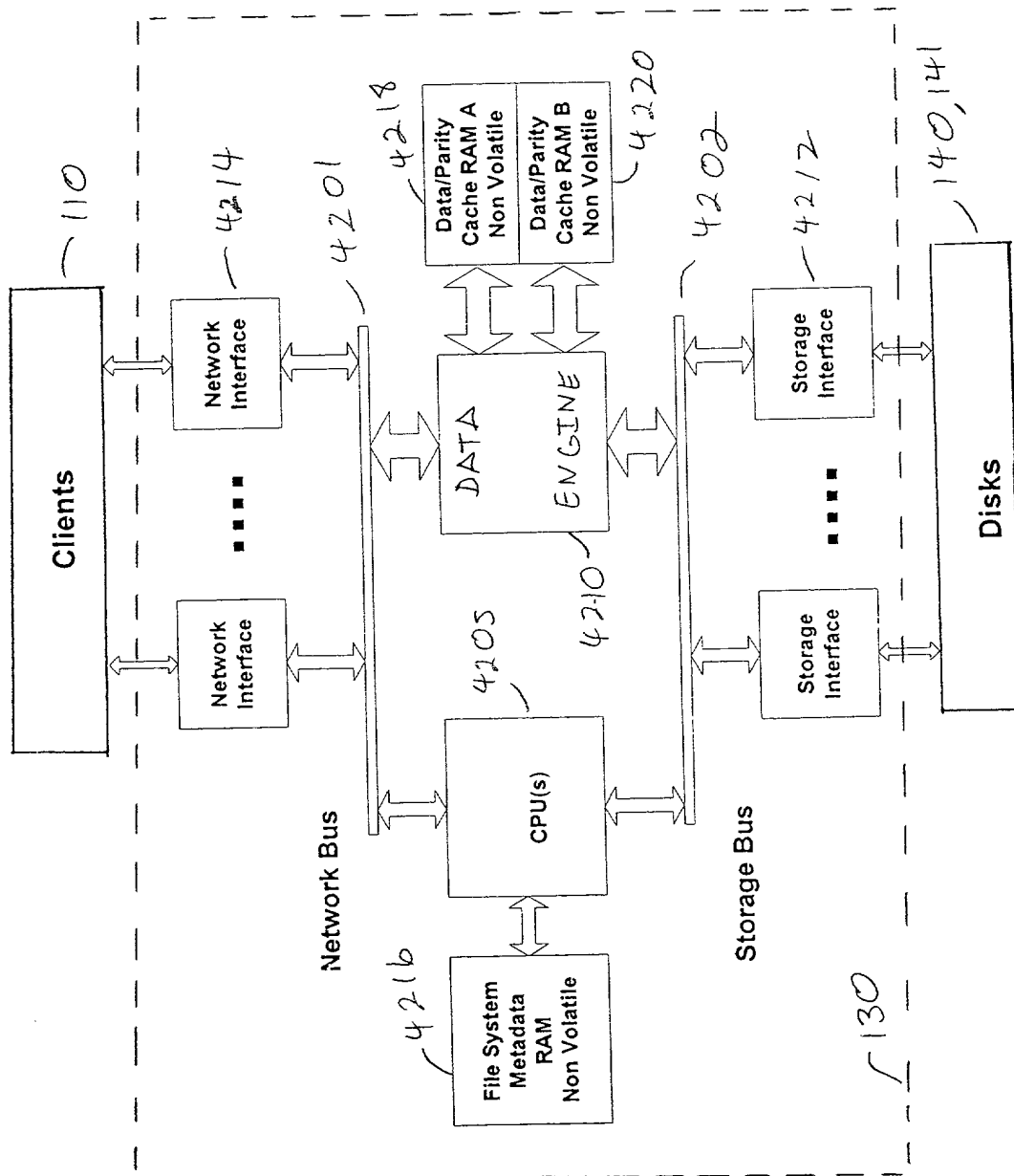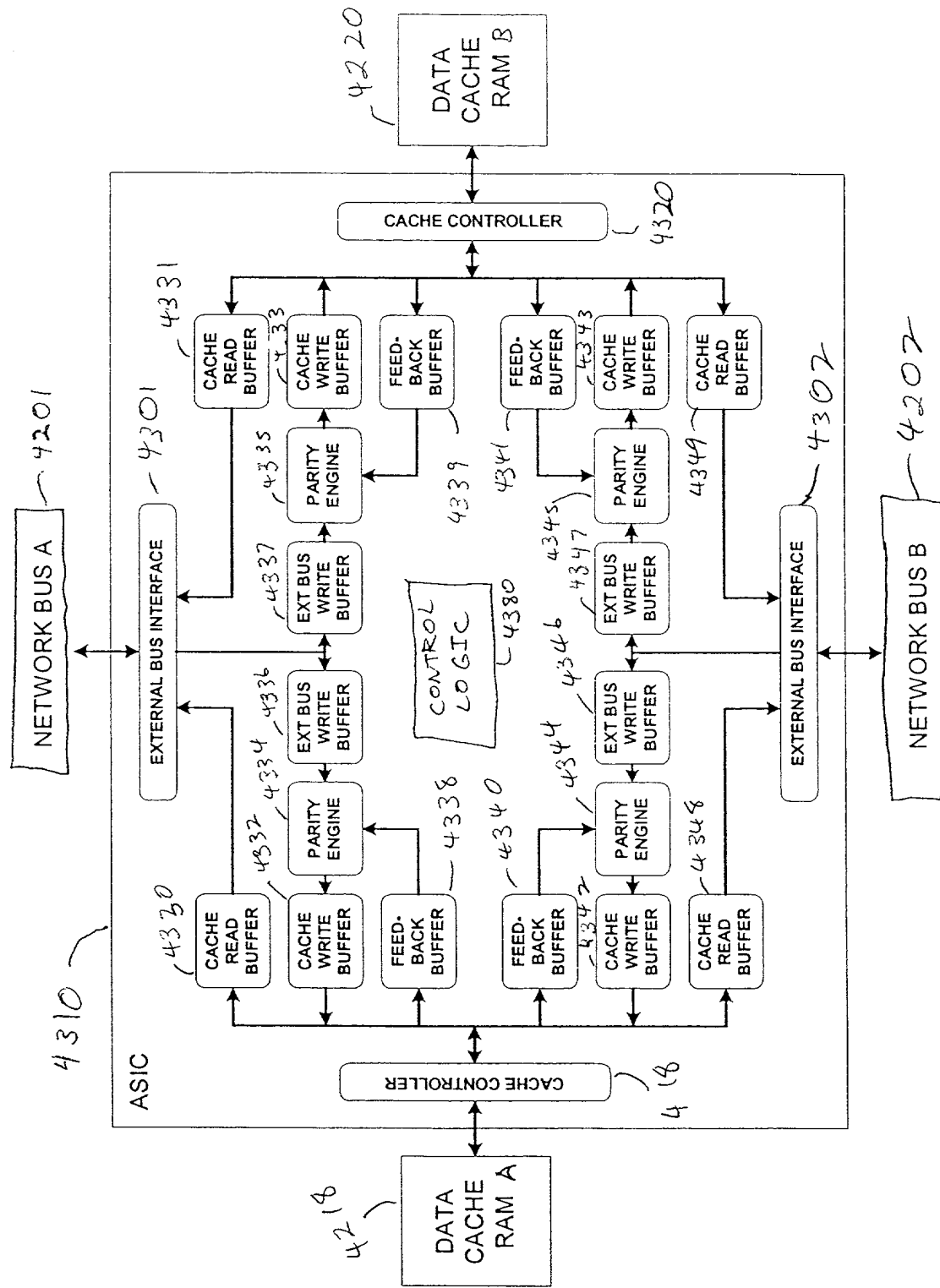 -Send file handle of SF to F1

4025

FIG. 41

Clients — 110

Network Interface — 4214

4201

Network Interface

Network Bus

CPU(s) — 4205

File System Metadata RAM Non Volatile — 4216

DATA ENGINE — 4210

Data/Parity Cache RAM A Non Volatile — 4218

Data/Parity Cache RAM B Non Volatile — 4220

4203

Storage Interface — 4212

Storage Interface

Storage Bus

Disks — 140,141

130

FIGURE 42

FIGURE 43

PCI map	Block Size	Opcode	Spare	Parity Index	Spare	RAM Adr

63----62,61-----59,58------56,55------51,50----35,34,32,31----------0

4400

FIGURE 44